01 Jan 2007

Online Reinforcement Learning Control of Unknown Nonaffine Nonlinear Discrete Time Systems

Qinmin Yang

Jagannathan Sarangapani
Missouri University of Science and Technology, sarangap@mst.edu

Recommended Citation

Q. Yang and J. Sarangapani, "Online Reinforcement Learning Control of Unknown Nonaffine Nonlinear Discrete Time Systems," *Proceedings of the 46th IEEE Conference on Decision and Control*, Institute of Electrical and Electronics Engineers (IEEE), Jan 2007.
The definitive version is available at https://doi.org/10.1109/CDC.2007.4434959

Proceedings of the
46th IEEE Conference on Decision and Control
New Orleans, LA, USA, Dec. 12-14, 2007

Online Reinforcement Learning Control of Unknown Nonaffine Nonlinear Discrete Time Systems

Qinmin Yang and S. Jagannathan

Abstract—**In this paper, a novel neural network (NN) based online reinforcement learning controller is designed for nonaffine nonlinear discrete-time systems with bounded disturbances. The nonaffine systems are represented by nonlinear auto regressive moving average with eXogenous input (NARMAX) model with unknown nonlinear functions. An equivalent affine-like representation for the tracking error dynamics is developed first from the original nonaffine system. Subsequently, a reinforcement learning-based neural network (NN) controller is proposed for the affine-like nonlinear error dynamic system. The control scheme consists of two NNs. One NN is designated as the critic, which approximates a predefined long-term cost function, whereas an action NN is employed to derive a control signal for the system to track a desired trajectory while minimizing the cost function simultaneously. Offline NN training is not required and online NN weight tuning rules are derived. By using the standard Lyapunov approach, the uniformly ultimate boundedness (UUB) of the tracking error and weight estimates is demonstrated.**

I. INTRODUCTION

Past literature [5]-[8] has reported the design of adaptive NN controllers for affine nonlinear discrete-time systems. However, for an unknown nonaffine nonlinear discrete-time system, such controller techniques cannot be directly employed. Further, reinforcement learning control techniques, which offer near optimal control solutions for nonaffine nonlinear discrete-time systems is not available except for affine systems [8].

One of the most popularly used nonaffine nonlinear discrete-time representation is nonlinear autoregressive moving average with eXogenous input (NARMAX) model, which is studied in [1]-[3]. Due to the difficulty in developing the controller design for nonaffine nonlinear discrete-time systems, an affine-like representation is first obtained and subsequently a controller is designed [1]-[3]. However, certain stringent assumptions are exerted, e.g. boundedness of control input changes, which limits its applicability for many practical applications. Moreover, reinforcement learning is no utilized and disturbances are not considered. In this paper, an affine-like representation is first derived by using Mean Value Theorem from the original NARMAX model and by relaxing the stringent assumption on the input changes [2]-[3]. Furthermore, to the best of our knowledge, so far no researcher has addressed the controller optimality for nonaffine nonlinear discrete-time systems, which is the main motivation of this effort.

Dynamic programming (DP) is a widely used methodology for solving optimization problems over time [4]. However, DP methods encounter the problem of "curse of dimensionality". Further, the DP schemes require that a) the nonlinear system under consideration is time-invariant, and b) large number of offline trials can be performed for the controller to approach optimality, which are usually not practical in real-time control.

To overcome the problems above, several appealing online neural network (NN) designs were introduced in [4]-[6], which were also referred to as forward dynamic programming (FDP) or adaptive critic designs (ACD). The central theme of this family of methods is that the optimal control law and cost function are approximated by parametric structures, such as NNs, which are trained over time along with the feedback from the plant. In other words, instead of finding the exact minimum, the ACDs approximate the Bellman equation: $J(x(k)) = \min_{u(k)} \{J(x(k+1)) + U(x(k), u(k))\}$, where $x(k)$ is the state and $u(k) = u(x(k))$ is a control law at time step k. The strategic utility function $J(x(k), u) = J(x(k))$ represents the cost or performance measure associated with going from k to final step N, while $U(x(k), x(k+1))$ is the utility function denoting the cost incurred in going from $x(k)$ to $x(k+1)$ by using control $u(k)$.

A new NN learning algorithm based on gradient descent rule is introduced in [7] for ACD design by using a simplified binary reward or cost function. The work in [8] proposes a near optimal controller design using standard Bellman equation for general affine nonlinear discrete-time systems.

In this paper, we are considering NNs for the control of nonaffine nonlinear discrete systems with quadratic-performance index as the cost function. The entire closed-loop system consists of two NNs: an action NN to derive the optimal (or near optimal) control signal to track not only the desired system output but also to minimize the long-term cost function; an adaptive critic NN to approximate the long-term cost function $J(x(k))$ and to tune the action NN weights. Closed-loop stability is demonstrated using Lyapunov.

II. BACKGROUND

A. System Dynamics

Consider the following non-affine discrete-time system

The authors are with the Department of Electrical & Computer Engineering, University of Missouri-Rolla, MO, 65401 USA (e-mail: qyy74@ umr.edu). Research supported in part by NSF grants ECCS#0327877, ECCS#0621924 and Intelligent Systems Center.

with disturbance given in NARMAX form [1] by

$$y(k+\tau)=f(\bar{y}_k,\bar{u}_{k-1},u(k),\bar{d}_{k+\tau-1})=f(w_k,u(k),\bar{d}_{k+\tau-1}) \quad (1)$$

where $w_k=\left[\bar{y}_k^T,\bar{u}_{k-1}^T\right]^T$, $\bar{y}_k=\left[y(k),...,y(k-n+1)\right]^T$, $\bar{u}_{k-1}=\left[u(k-1),...,y(k-n+1)\right]^T$ denotes the system outputs and inputs respectively. The term $\bar{d}_{k+\tau-1}=\left[d(k+\tau-1),...,d(k)\right]^T$ is the disturbance, and τ represents the system delay, or the relative degree of the system [2]. Note that the output $y(k)$ is considered measurable with initial values in a compact set Ω_{y_0}.Furthermore, several mild assumptions are needed in order to proceed.

Assumption 1: The unknown nonlinear function $f(\cdot)$ in (1) is continuous and differentiable.

Assumption 2: The disturbance $d(k)$ is bounded with a known bound $|d(k)|\le d_M$, and the partial derivative $|\partial f/\partial d(k)|\le D_M$ is also bounded, with D_M a positive constant.

With assumption 2, by using Mean Value Theorem, equation (1) can be rephrased as

$$\begin{aligned}y(k+\tau)&=f(w_k,u(k),\bar{d}_{k+\tau-1})=f(w_k,u(k),0)+\delta_f^T\bar{d}_{k+\tau-1}\\&=f(w_k,u(k),0)+\delta_{d_k}\end{aligned} \quad (2)$$

where $\delta_f=\left[\left.\frac{\partial f}{\partial d(k+\tau-1)}\right|_{d(k+\tau-1)=d_\xi(k+\tau-1)},...,\left.\frac{\partial f}{\partial d(k)}\right|_{d(k)=d_\xi(k)}\right]^T$, $\delta_{d_k}=\delta_f^T\bar{d}_{k+\tau-1}$,

and $d_\xi(k)$ is between 0 and $d(k)$, or $d_\xi(k)=0+\lambda(d(k)-0)$, $\lambda\in[0,1]$. Through this paper, they have the same meaning, and we will present by the former fashion for simplicity.

Lemma 1: δ_{d_k} is bounded by $|\delta_{d_k}|\le\tau D_M d_M$.

Proof: Use (2) and Assumption 2.

Our objective is to design a control law to drive the system output $y(k)$ to track a desired trajectory $y_d(k)$. Before we proceed, let us construct the following virtual system, which is free of disturbance.

$$y_n(k+\tau)=f(w_k,u(k),0) \quad (3)$$

Assumption 3: $\partial f/\partial u(k)=g(k)$ is bounded and satisfies $0<g_{\min}\le g(k)\le g_{max}$, where $g_{\min}$ and $g_{\max}$ are positive constants [8].

Assumption 4: Virtual system (3) is invertibly stable [9], which means bounded system output implies bounded system input.

From Assumptions 3 and 4, we can draw the conclusion that for any output trajectory $y_n(k+\tau)=f(w_k,u(k),0)$, there exists a unique and continuous (smooth) function $u(k)=f^{-1}(w_k,y_n(k+\tau),0)$ [2], [11].

B. Optimal Control

In this paper, we consider the long-term cost function as

$$\begin{aligned}J(k)&=J(y(k),u)=\sum_{i=t_0}^{\infty}\gamma^i r(k+i)\\&=\sum_{i=t_0}^{\infty}\gamma^i[q(y(k+i))+u^T(k+i)Ru(k+i)]\end{aligned} \quad (4)$$

where $J(k)$ stands for $J(x(k),u)$ for simplicity, u is a given control policy, R is a positive design constant and γ $(0\le\gamma\le1)$ is the discount factor for the infinite-horizon problem [8]. One can observe from (4) that the long-term cost function is the discounted sum of the immediate cost function or Lagrangian expressed as

$$\begin{aligned}r(k)&=q(y(k))+u^T(k)Ru(k)\\&=(y(k)-y_d(k))^TQ(y(k)-y_d(k))+u^T(k)Ru(k)\\&=Qe^2(k)+Ru^2(k)\end{aligned} \quad (5)$$

where Q is a positive design constant. In this paper, we are using a widely applied standard quadratic cost function defined based on the tracking error $e(k)=y(k)-y_d(k)$ in contrast with [7]. The immediate cost function $r(k)$ can be viewed as the system performance index for the current step.

The basic idea in the adaptive critic or reinforcement learning design is to approximate the long-term cost function $J(k)$ (or its derivative, or both), and generate the control signal minimizing the cost. By online learning phenomenon, the online approximator will converge to the optimal cost function and the controller will in turn generate an optimal signal. As a matter of fact, for an optimal control law, which can be expressed as $u^*(k)=u^*(y(k))$, the optimal long-term cost function can be written as $J^*(k)=J^*(y(k),u^*(y(k)))=J^*(y(k))$, which is just a function of the current system output [10]. Next, one can state the following assumption.

Assumption 5: The optimal cost function $J^*(k)$ is finite and bounded over the compact set $S\subset R$ by J_M.

III. AFFINE-LIKE DYNAMICS

Next an affine like representation is derived by applying the Taylor series expansion of system (3) with respect to $u(k)$ around $u(k-1)$ yields

$$\begin{aligned}y(k+\tau)&=f(w_k,u(k),0)+\delta_{d_k}\\&=f(w_k,u(k-1),0)+\frac{\partial f(w_k,u(k-1),0)}{\partial u}\Delta u(k)\\&\quad+\frac{1}{2}\cdot\frac{\partial^2 f(w_k,u_\mu(k),0)}{\partial u^2}\Delta u^2(k)+\delta_{d_k}\\&=F(w_k,u(k))+G(w_k)\Delta u(k)+\delta_{d_k}\end{aligned} \quad (5)$$

where

$$F(w_k,u(k))=f(w_k,u(k-1),0)+\frac{1}{2}\cdot\frac{\partial^2 f(w_k,u_\mu(k),0)}{\partial u^2}\Delta u^2(k)$$

$$G(w_k)=\frac{\partial f(w_k,u(k-1),0)}{\partial u}$$

where $u_\mu(k)$ is between $u(k)$ and $u(k+1)$ (or

$u_\mu(k) = u(k+1) + \lambda(u(k+1) - u(k)), \lambda \in [0,1]$) by using Mean Value Theorem. In other words, there are no higher order terms in the Taylor series expansion missing, since they are incorporated into the second derivative. By observing (5), we have the equation similar to the virtual system as

$$y_n(k+\tau) = F(w_k, u(k)) + G(w_k)\Delta u(k) \tag{6}$$

Lemma 2: Consider any desired system trajectory $y_d(k) \in R$ and corresponding nominal desired control input $u_d(k) = f^{-1}(w_k, y_d(k+\tau), 0)$, there exists $u_\xi(k)$ between any input $u_n(k)$ and $u_d(k)$ to the system such that

$$\begin{aligned} F(w_k, u_n(k)) = F(w_k, u_d(k)) + \frac{\partial F(w_k, u_\xi(k))}{\partial u} \\ \times \frac{\partial f^{-1}(w_k, y_\xi(k+\tau), 0)}{\partial y} \cdot (y_n(k+\tau) - y_d(k+\tau)) \end{aligned} \tag{7}$$

where $u_\xi(k) = f^{-1}(w_k, y_\xi(k+\tau), 0)$.

Lemma 3: Consider the output of the virtual system $y_n(k+\tau) = f(w_k, u_n(k), 0)$ for a given input $u_n(k)$, then there exists $y_\varsigma(k+\tau)$ between $y_n(k+\tau)$ and $y_d(k+\tau)$ such that

$$\begin{aligned} u_n(k) &= f^{-1}(w_k, y_n(k+\tau), 0) \\ &= f^{-1}(w_k, y_d(k+\tau), 0) + \frac{\partial f^{-1}(w_k, y_\varsigma(k+\tau), 0)}{\partial y}(y_n(k+\tau) - y_d(k+\tau)) \\ &= u_d(k) + \frac{\partial f^{-1}(w_k, y_\varsigma(k+\tau), 0)}{\partial y}(y_n(k+\tau) - y_d(k+\tau)) \end{aligned} \tag{8}$$

Proof: Lemmas 2 and 3 can be obtained by using Chain Rule and Mean Value Theorem. Further, we have following lemma.

Lemma 4: Consider system (6) with Lemma 2 and 3, we have

$$\frac{\partial F(w_k, u_\xi(k))}{\partial u} \cdot \frac{\partial f^{-1}(w_k, y_\xi(k+\tau), 0)}{\partial y} + G(w_k)\frac{\partial f^{-1}(w_k, y_\varsigma(k+\tau), 0)}{\partial y} = 1 \tag{9}$$

Proof: i) If $y_n(k+\tau) = y_d(k+\tau)$, then $y_\xi(k+\tau) = y_\varsigma(k+\tau) = y_d(k+\tau)$. Therefore, (9) could be obtained by differentiating (6) with respect to $y_n(k+\tau)$.

ii) If $y_n(k+\tau) \neq y_d(k+\tau)$, then from (6), one has

$$\begin{aligned} y_n(k+\tau) &= F(w_k, u_n(k)) + G(w_k)(u_n(k) - u(k-1)) \\ &= F(w_k, u_d(k)) + \frac{\partial F(w_k, u_\xi(k))}{\partial u} \\ &\times \frac{\partial f^{-1}(w_k, y_\xi(k+\tau), 0)}{\partial y} \cdot (y_n(k+\tau) - y_d(k+\tau)) \\ &+ G(w_k)(u_d(k) - u(k-1)) \\ &+ \frac{\partial f^{-1}(w_k, y_\varsigma(k+\tau), 0)}{\partial y}(y_n(k+\tau) - y_d(k+\tau)) \\ &= F(w_k, u_d(k)) + G(w_k)(u_d(k) - u(k-1)) \\ &+ (\frac{\partial F(w_k, u_\xi(k))}{\partial u}\frac{\partial f^{-1}(w_k, y_\xi(k+\tau), 0)}{\partial y} \\ &+ \frac{\partial f^{-1}(w_k, y_\varsigma(k+\tau), 0)}{\partial y}) \times (y_n(k+\tau) - y_d(k+\tau)) \end{aligned} \tag{10}$$

Substituting $y_d(k+\tau) = F(w_k, u_d(k)) + G(w_k)(u_d(k) - u(k-1))$ into (10) yields

$$\frac{\partial F(w_k, u_\xi(k))}{\partial u} \cdot \frac{\partial f^{-1}(w_k, y_\xi(k+\tau), 0)}{\partial y} + G(w_k)\frac{\partial f^{-1}(w_k, y_\varsigma(k+\tau))}{\partial y} = 1$$

Lemma 5: For any $y_\varsigma(k+\tau) \in S$ and corresponding control input $u_\varsigma(k) = f^{-1}(w_k, y_\varsigma(k+\tau), 0)$, the following statement holds

$$\frac{\partial f(w_k, u_\varsigma(k), 0)}{\partial u} \cdot \frac{\partial f^{-1}(w_k, y_\varsigma(k+\tau))}{\partial y} = 1 \tag{11}$$

Proof: It can be straightforward to verify (11) by differentiating $y_\varsigma(k+\tau) = f(w_k, f^{-1}(w_k, y_\varsigma(k+\tau)), 0)$ with respect to $y_\varsigma(k+\tau)$.

Therefore, substituting (7) into (5) produces the system dynamics in terms of the tracking error as

$$\begin{aligned} e(k+\tau) &= y(k+\tau) - y_d(k+\tau) \\ &= F(w_k, u(k)) + G(w_k)\Delta u(k) + \delta_{d_k} - y_d(k+\tau) \\ &= F(w_k, u_d(k)) + \frac{\partial F(w_k, u_\xi(k))}{\partial u} \\ &\times \frac{\partial f^{-1}(w_k, y_\xi(k+\tau), 0)}{\partial y} \cdot (y(k+\tau) - y_d(k+\tau)) \\ &+ G(w_k)\Delta u(k) + \delta_{d_k} - y_d(k+\tau) \end{aligned} \tag{12}$$

Making use of Lemma 4, (12) can be written as

$$\begin{aligned} e(k+\tau) &= F(w_k, u_d(k)) + G(w_k)\Delta u(k) + \delta_{d_k} - y_d(k+\tau) \\ &+ (1 - G(w_k)\frac{\partial f^{-1}(w_k, y_\varsigma(k+\tau), 0)}{\partial y}) \cdot (y(k+\tau) - y_d(k+\tau)) \\ &= F(w_k, u_d(k)) + G(w_k)\Delta u(k) + \delta_{d_k} - y_d(k+\tau) \\ &+ (1 - G(w_k)\frac{\partial f^{-1}(w_k, y_\varsigma(k+\tau), 0)}{\partial y}) \cdot e(k+\tau) \end{aligned} \tag{13}$$

Combing (11) and (13), one has

$$e(k+\tau) = \frac{\partial f(w_k, u_\varsigma(k), 0)}{\partial u}(\frac{F(w_k, u_d(k)) + \delta_{d_k} - y_d(k+\tau)}{G(w_k)} + \Delta u(k)) \tag{14}$$

By defining $\partial f(w_k, u_\varsigma(k), 0)/\partial u = \kappa_k$, (14) can be rephrased as

$$\begin{aligned} e(k+\tau) &= \frac{\kappa_k}{G(w_k)}(F(w_k, u_d(k)) - y_d(k+\tau)) + \kappa_k \Delta u(k) + \frac{\kappa_k}{G(w_k)}\delta_{d_k} \\ &= F_a(w_k, y_d(k+\tau), \kappa_k) + \kappa_k \Delta u(k) + \delta_{\kappa_k} \end{aligned} \tag{15}$$

$$F_a(w_k, y_d(k+\tau), \kappa_k) = \frac{\kappa_k}{G(w_k)}(F(w_k, u_d(k)) - y_d(k+\tau))$$

$$\delta_{\kappa_k} = \frac{\kappa_k}{G(w_k)}\delta_{d_k}$$

Notice that $0 < g_{\min} \le \kappa_k \le g_{max}, 0 < g_{\min} \le G(w_k) \le g_{max}$ due to Assumption 3. By referring to Lemma 1, one also observes that δ_{κ_k} is bounded above by $|\delta_{\kappa_k}| \le g_{max}\tau D_M d_M / g_{\min}$.

By rewriting the non-affine system into an equivalent affine-like representation (15) in terms of error dynamics, the difficulty of designing controllers for nonaffine nonlinear

discrete-time systems could be avoided.

IV. Online Reinforcement Learning Controller Design

The purpose of this study is to design an online reinforcement learning NN controller for the equivalent error dynamics (15), such that 1) all the signals in the closed-loop system remain UUB; 2) the output $y(k)$ follows a desired trajectory $y_d(k) \in S$; and 3) the long-term cost function (4) is minimized so that a near optimal control input can be generated [8]. Here, the "online" means the learning of the controller takes place "in real-time" by interacting with the plant, instead of in an offline or iterative manner.

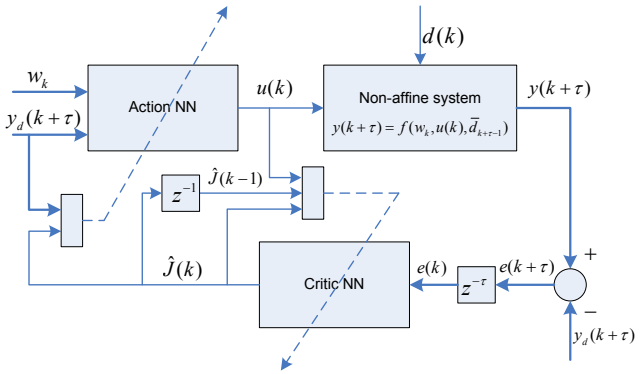

Fig. 1. Online reinforcement learning neural controller structure.

The block diagram of the proposed controller is shown in Fig. 1, where the two NN weights are initialized to zero and trained online without any offline learning phase.

In our controller architecture, we consider the action and the critic NN having two layers, and the output of the NN can be given by $Y = W^T \phi(V^T X)$, where V and W are the hidden layer and output layer weights respectively. X is the input vector of the NN and we choose $\phi(X) = 2/(1+e^{-X}) - 1$ as the activation function.

We know that any continuous function $f(X) \in C^N(S)$ can be written as

$$f(X) = W^T \phi(V^T X) + \varepsilon(X) \tag{16}$$

with $\varepsilon(X)$ a NN functional reconstruction error vector. In our design, W is adapted online but V is initially selected at random and held fixed [11]. Furthermore, in this paper, a novel tuning algorithm is proposed making the NN weights robust by relaxing the persistency of excitation (PE) condition unnecessary. Next, the following mild assumption is needed.

Assumption 6: The desired trajectory of the system output, $y_d(k)$, is bounded over the compact subset of R.

A. The Action NN Design

Consider system (15), to eliminate the tracking error, a desired control law is given by

$$u_d(k) = u(k-1) - \frac{1}{\kappa_k} F_a(w_k, y_d(k+\tau), \kappa_k) \tag{17}$$

By this means, the tracking error will go to zero after τ steps if no disturbance presents.

However, since both of $F_a(w_k, y_d(k+\tau), \kappa_k)$ and κ_k are unknown smooth nonlinear functions, the desired feedback control $u_d(k)$ cannot be implemented directly. Instead, an action NN is employed to produce the control signal. From (17) and considering Assumption 3 and 4, the desired control signal can be approximated by the action NN as

$$u_d(k) = w_a^T \phi_a(v_a^T s(k)) + \varepsilon_a(s(k)) = w_a^T \phi_a(k) + \varepsilon_a(k) \tag{18}$$

where $s(k) = \left[w_k^T, y_d(k+\tau)\right]^T$ is the action NN input vector. n_a is the number of neurons in the hidden layer, and $w_a \in R^{n_a \times 1}$, $v_a \in R^{2n \times n_a}$ denote the desired weights of the output and hidden layer respectively with $\varepsilon_a(k) = \varepsilon_a(s(k))$ is the action NN approximation error. Since v_a is fixed, for simplicity purpose, the hidden layer activation function vector $\phi_a(v_a^T s(k)) \in R^{n_a}$ is written as $\phi_a(k)$.

Considering the fact that the desired weights are unavailable for us, the actual NN weights have to be trained online and its actual output can be expressed as

$$u(k) = \hat{w}_a^T(k) \phi_a(v_a^T s(k)) = \hat{w}_a^T(k) \phi_a(k) \tag{19}$$

where $\hat{w}_a(k) \in R^{n_a \times 1}$ is the actual weight matrix of the output layer at instant k.

Using the action NN output as the control signal, and substituting (18) and (19) into (15) yields

$$\begin{aligned} e(k+\tau) &= F_a(w_k, y_d(k+\tau), \kappa_k) + \kappa_k \Delta u(k) + \delta_{\kappa_k} \\ &= \kappa_k (u(k) - u_d(k)) + \delta_{\kappa_k} \\ &= \kappa_k (\tilde{w}_a^T(k) \phi_a(k) - \varepsilon_a(k)) + \delta_{\kappa_k} \\ &= \kappa_k \zeta_a(k) + d_a(k) \end{aligned} \tag{20}$$

where

$$\tilde{w}_a(k) = \hat{w}_a(k) - w_a \tag{21}$$

$$\zeta_a(k) = \tilde{w}_a^T(k) \phi_a(k) \tag{22}$$

$$d_a(k) = -\kappa_k \varepsilon_a(k) + \delta_{\kappa_k} \tag{23}$$

Next the critic NN design with updating rule is followed.

B. The Critic NN Design

As stated above, a near optimal controller can stabilize the closed-loop system along with minimizing the cost function. In this regard, a critic NN is employed to approximate the unknown long-term cost function $J(k)$ for current stage.

First, the prediction error generated by the critic or the Bellman error [9] is defined as

$$e_c(k) = \gamma \hat{J}(k) - \hat{J}(k-1) + r(k) \tag{24}$$

where the subscript "c" stands for the "critic" and

$$\hat{J}(k) = \hat{w}_c^T(k) \phi_c(v_c^T e(k)) = \hat{w}_c^T(k) \phi_c(k) \tag{25}$$

$\hat{J}(k) \in R$ is the critic NN output which is for approximating

$J(k)$. $\hat{w}_c(k) \in R^{n_c \times 1}$ and $v_c \in R^{1 \times n_c}$ represent the actual weight matrices of the output and hidden layer respectively. The term n_c denotes the number of the neurons in the hidden layer. Similar to HDP, the tracking error $e(k)$ are selected as the critic NN input. Again, the activation function vector of the hidden layer $\phi_c(v_c^T e(k)) \in R^{n_c}$ is simply denoted as $\phi_c(k)$. Provided with enough number of hidden layer neurons, the optimal long-term cost function $J^*(k)$ can be approximated with arbitrarily small approximation error $\varepsilon_c(k)$ as

$$J^*(k) = w_c^T \phi_c(v_c^T e(k)) + \varepsilon_c(e(k)) = w_c^T \phi_c(k) + \varepsilon_c(k) \quad (26)$$

Similarly, the critic NN weight estimation error can be defined as

$$\tilde{w}_c(k) = \hat{w}_c(k) - w_c \quad (27)$$

where the approximation error is given by

$$\zeta_c(k) = \tilde{w}_c^T(k)\phi_c(k) \quad (28)$$

Thus, we obtain

$$\begin{aligned} e_c(k) &= \gamma \hat{J}(k) - \hat{J}(k-1) + r(k) \\ &= \gamma \zeta_c(k) + \gamma J^*(k) - \zeta_c(k-1) - J^*(k-1) \\ &\quad + r(k) - \varepsilon_c(k) + \varepsilon_c(k-1) \end{aligned} \quad (29)$$

Next we propose the weight tuning algorithms for both NNs.

C. Weight Updating for the Critic NN

Following the discussion from the last section, the objective function to be minimized by the critic NN can be defined as a quadratic function of Bellman error as

$$E_c(k) = \frac{1}{2} e_c^T(k) e_c(k) = \frac{1}{2} e_c^2(k) \quad (30)$$

Using a standard gradient-based adaptation method, the weight updating algorithm for the critic NN is given by

$$\hat{w}_c(k+\tau) = \hat{w}_c(k) + \Delta \hat{w}_c(k) \quad (31)$$

where

$$\Delta \hat{w}_c(k) = \alpha_c \left[-\frac{\partial E_c(k)}{\partial \hat{w}_c(k)} \right] \quad (32)$$

with $\alpha_c \in R$ is the adaptation gain.

Combining (24), (25), (30) with (32), the critic NN weight updating rule can be obtained by using the chain rule as [8]

$$\begin{aligned} \Delta \hat{w}_c(k) &= -\alpha_c \frac{\partial E_c(k)}{\partial \hat{w}_c(k)} = -\alpha_c \frac{\partial E_c(k)}{\partial e_c(k)} \frac{\partial e_c(k)}{\partial \hat{J}(k)} \frac{\partial \hat{J}(k)}{\partial \hat{w}_c(k)} \\ &= -\alpha_c \gamma \phi_c(k)(\gamma \hat{J}(k) + r(k) - \hat{J}(k)) \end{aligned} \quad (33)$$

D. Weight Updating for the Action NN

The objective for adapting the action NN is to track the desired output and to lower the cost function simultaneously. Therefore, the error for the action NN can be formed by combining the functional estimation error $\zeta_a(k)$, and the critic signal $\hat{J}(k)$. Let

$$e_a(k) = \sqrt{\kappa_k}\zeta_a(k) + \left(\sqrt{\kappa_k}\right)^{-1}\left(\hat{J}(k) - J_d(k)\right) = \sqrt{\kappa_k}\zeta_a(k) + \left(\sqrt{\kappa_k}\right)^{-1}\hat{J}(k) \quad (34)$$

where $\zeta_a(k)$ is defined in (22). The desired long-term cost function $J_d(k)$ is nominally defined and is considered to be zero ("0"), which means as low as possible [8].

Hence, the weights of the action NN $\hat{w}_a(k)$ are tuned to minimize the error

$$E_a(k) = \frac{1}{2} e_a^T(k) e_a(k) \quad (35)$$

Combining (20), (22), (34), (35) and using the chain rule yields

$$\begin{aligned} \Delta \hat{w}_a(k) &= -\alpha_a \frac{\partial E_a(k)}{\partial \hat{w}_a(k)} = -\alpha_a \frac{\partial E_a(k)}{\partial e_a(k)} \frac{\partial e_a(k)}{\partial \zeta_a(k)} \frac{\partial \zeta_a(k)}{\partial \hat{w}_c(k)} \\ &= -\alpha_a \phi_a(k)(\kappa_k \zeta_a(k) + \hat{J}(k))^T \\ &= -\alpha_a \phi_a(k)(e(k+\tau) - d_a(k) + \hat{J}(k))^T \end{aligned} \quad (36)$$

where $\alpha_a \in R^+$ is the adaptation gain of the action NN. Since $d_a(k)$ is typically unavailable, similar to the ideal case, we assume the $d(k)$ and the mean value of $\varepsilon_a(k)$ over the compact subset of R to be zero [8], and obtain the weight updating algorithm for the action NN as

$$\hat{w}_a(k+\tau) = \hat{w}_a(k) - \alpha_a \phi_a(k)(e(k+\tau) + \hat{J}(k))^T \quad (37)$$

V. Main Theoretic Result

Assumption 7: Let the unknown desired output layer weights for the action and critic NNs are upper bounded such that

$$\|w_a\| \le w_{am}, \text{ and } \|w_c\| \le w_{cm} \quad (38)$$

where $w_{am} \in R^+$ and $w_{cm} \in R^+$ represent the bounds on the unknown target weights. Here $\|\cdot\|$ stands for the Frobenius norm [14].

Assumption 8: The activation functions for the action and critic NNs are bounded by known positive values, such that

$$\|\phi_a(k)\| \le \phi_{am}, \ \|\phi_c(k)\| \le \phi_{cm} \quad (39)$$

where $\phi_{am}, \phi_{cm} \in R^+$ is the upper bound. It is easily satisfied, since hyperbolic tangent sigmoid transfer function is chosen.

Assumption 9: The NN approximation errors or unmodeled dynamics $\varepsilon_a(k)$ and $\varepsilon_c(k)$ are assumed to be bounded above over the compact set $S \subset R$ by ε_{am} and ε_{cm} [10].

Lemma 6: With the Assumption 3, 9, the term $d_a(k)$ in (23) is bounded over the compact set $S \subset R$ by

$$|d_a(k)| \le d_{am} = g_{\max}\varepsilon_{am} + g_{max}\tau D_M d_M / g_{\min} \quad (40)$$

Combining Assumptions 1, 3, and 4 and Facts 1 and 2, the main result is introduced next.

Theorem 1: Consider the nonlinear discrete-time system given by (1) whose dynamics can be expressed as (15). Let the Assumptions 1 through 9 hold with the disturbance bound d_M a known constant. Let the control input be provided by the action NN (19), with the critic NN (25). Further, let the weights of the action NN and the critic NN be tuned by (33)

and (37) respectively. Then the tracking error $e(k)$, and the NN weight estimates of the action and critic NNs, $\tilde{w}_a(k)$ and $\tilde{w}_c(k)$ are *UUB* [12], provided the controller design parameters are selected as

(a) $$0 < \alpha_a \phi_a^2(k) < \frac{g_{\min}}{g_{\max}^2} \quad (41)$$

(b) $$0 < \alpha_c \phi_c^2(k) < 1/\gamma^2 \quad (42)$$

(c) $$\gamma > \frac{1}{2} \quad (43)$$

where α_a and α_c are NN adaptation gains, and γ is employed to define the *strategic* utility function.

Proof: the proof is omitted here due to space limitation.

Remark: Compared to other adaptive critic or reinforcement learning schemes [4]-[7], the proposed approach ensures closed-loop stability using the Lyapunov approach even though gradient based adaptation is employed.

VI. Simulation Results

The system under consideration is governed by

$$y(k+1) = \sin(y(k)) + u(k)(5 + \cos(y(k)u(k))) + d \quad (44)$$

The desired output trajectory is set to be a rectangular pulse wave with amplitude of 2 units and period of 8s. The time interval between two instances is 0.02s and a bounded uniformly distributed noise d with bound of d_M is introduced into the simulation. Other parameters are listed as:

TABLE 1
SUMMARY OF PARAMETERS USED IN SIMULATION OF THE FIRST EXAMPLE

R	Q	γ	α_c	α_a	n_a	n_c	d_M
0.1	0.1	0.8	0.01	0.01	10	10	0.01

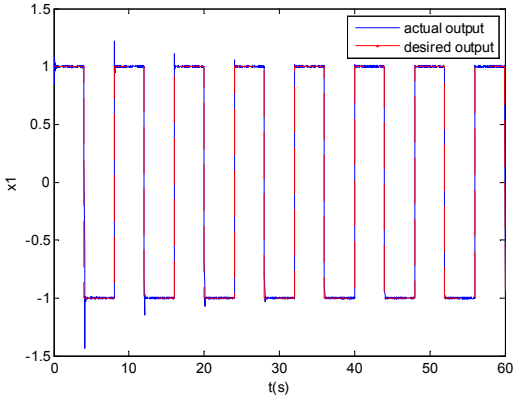

Fig. 2. Tracking performance of the online learning controller.

The tracking performance with the online reinforcement learning controller is shown in Fig. 2, which demonstrates that the actual system output is able to track the target quickly even under the influence of noise. Meanwhile, the control signal is shown in Fig. 3. From these results, the boundedness of the control input is clearly illustrated.

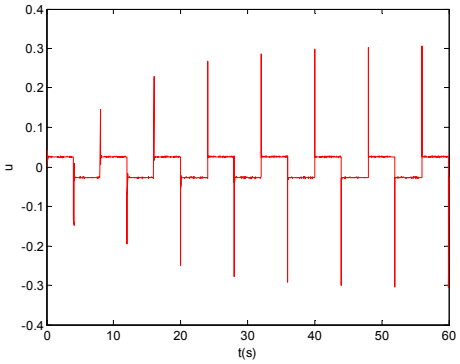

Fig. 3. Input signal of the online learning controller.

VII. Conclusions

A method to obtain equivalent affine-like model from nonlinear unknown nonaffine nonlinear discrete-time system is introduced in this paper. The model integrates all higher order terms of the Taylor expansion without losing any terms. Bounded disturbance is also considered. Subsequently, a novel reinforcement online learning scheme is designed to deliver a desired performance by using neural networks. The proposed NN controller optimizes the long-term cost function. To suit practical applications, the controller is updated in an online fashion without offline phase. The UUB of the closed-loop tracking errors and NN weight estimates is demonstrated by using standard Lyapunov analysis.

References

[1] S. A. Billings and W. S. F. Voon, "A prediction-error and stepwise regression algorithm for nonlinear systems," in International Journal of Control 44, pp. 803–822, 1986.

[2] O. Adetona, E. Garcia, and L. H. Keel, "A new method for the control of discrete nonlinear dynamic systems using neural networks," IEEE Trans. Neural Networks, vol. 11, pp. 102–112, Jan. 2000.

[3] O. Adetona, S. Sathananthan, and L. H. Keel, "Robust adaptive control of nonaffine nonlinear plants with small input signal changes," IEEE Trans. Neural Networks, vol. 15, pp. 408–416, Mar. 2004.

[4] J. Si, A. G. Barto, W. B. Powell, and D. Wunsch, Eds., "Handbook of Learning and App. Dynamic Programming", Wiley-IEEE Press, 2004.

[5] D. Prokhorov and D. Wunsch, "Adaptive critic designs", IEEE Trans. Neural Networks, Vol. 8, No.5, p.997-1007, 1997.

[6] P. J. Werbos, "Building and understanding adaptive systems: A statistical/numerical approach to factory automation and brain research," IEEE Transactions on Systems, Man, and Cybernetics 17, pp. 7–20, 1987.

[7] J. Si and Y. T. Wang, "On-line learning control by association and reinforcement," IEEE Trans. Neural Networks, vol. 12, no. 2, pp. 264–276, Mar.2001.

[8] Qinmin Yang and S. Jagannathan, "Online reinforcement learning-based neural network controller design for affine nonlinear discrete-time systems", Proc. of American Control Conference, 2007.

[9] M. S. Ahmed, "Neural-net-based direct adaptive control for a class of nonlinear plants," IEEE Trans. On Automatic Control, vol. 45, pp. 119–124, 2000.

[10] D. P. Bertsekas, "Dynamic Programming and Optimal Control. Belmont," MA: Athena Scientific, 2000.

[11] B. Igelnik and Y. H. Pao, "Stochastic choice of basis functions in adaptive function approximation and the functional-link net," IEEE Trans. Neural Network, vol. 6, no. 6, pp. 1320–1329, Nov. 1995.

[12] S. Jagannathan, "Neural Network Control of Nonlinear Discrete-time Systems", Taylor and Francis (CRC Press), Boca Raton, FL 2006.